

Parallel Computing for Science and Engineering

General home work instructions

3/9/2023

Instructor:

Lars Koesterke

General Home Work Guidelines

What you have to do!

- In your home directory:
- Create a directory pcse with subdirectories: **hw0** , **hw1** , ..., **ex1** , ...
 - Place your “final products” in these directories
 - Expect us to do either:
 - `cd hw1; make clean; make hw1`
 - Or to follow a `read.me` file
- Please print the results of your home work out!

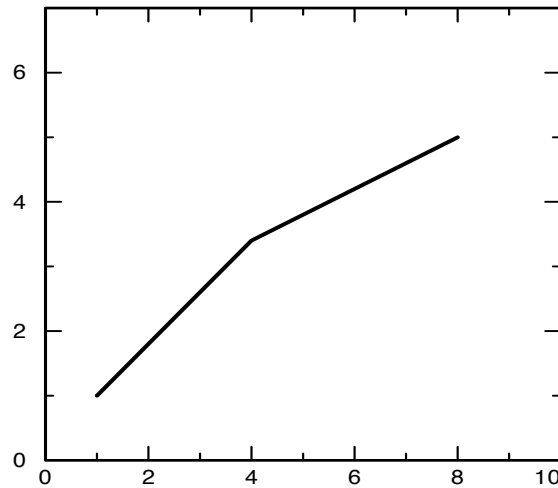
General Remarks

- Resource: Frontera
 - Preventive maintenance, typically Tuesdays (but not every Tuesday)
- Resource may face an emergency and may be inaccessible at any time.
- Also consider that you may have to wait in the queue
- Do not wait with your home work to the last minute!
- Let us know if you cannot make the deadline

What's wrong with this?

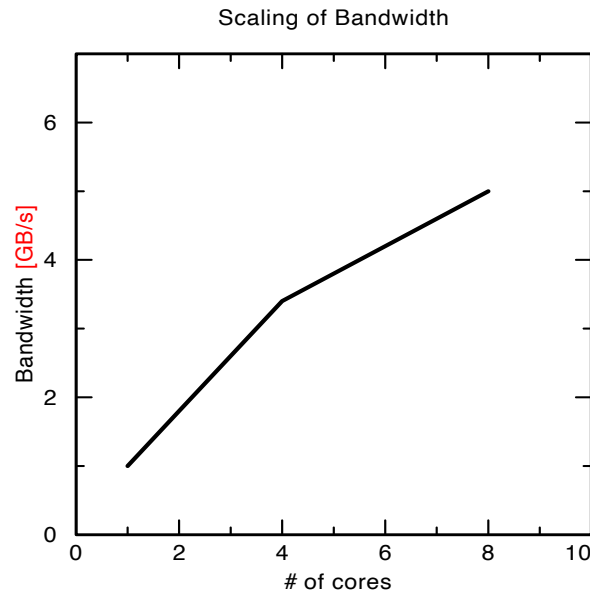
Let's get a bit organized

What's wrong with this?



What's wrong with this?

- A figure (or table) needs a caption
- The x and y-axis have to be labeled
- The labels need units, if applicable



What's wrong with this?

- The experiment ran 3 times. The results and the average are:

	#1	#2	#3
Run-time	17.612345	17.167890	18.112345
Average	17.63086		

What's wrong with this?

- Units are missing
- The scatter is about 0.5 seconds
- Timings below 0.01 seconds are apparently meaningless
- Apply appropriate rounding!

	#1	#2	#3
Run-time [s]	17.61	17.17	18.11
Average [s]	17.63		

How to get around in Linux

- Editors
 - vi (vim) or emacs
- Basic commands
 - cd, pwd, ls, rm, mv, cat, more, mkdir, rmdir, hostname, top, echo
 - ps, kill (later to identify and kill run-away processes)
 - gcc and gfortran for compiling
- Man pages
 - Example: man ls
- Logon to a remote resource with ssh (see Ian's intro)
- Google it!
 - Example: google for 'linux explain ls'
 - Look for tutorials
- Ask a friend, a co-worker, a fellow student
- Ask us

How to get around on a TACC System Frontera

- Logging on to Frontera
 - ssh, two-factor authentication, Putty for Windows users
- Basic commands
 - cd, pwd, ls, rm, mv, cat, more, mkdir, rmdir, hostname, top, echo
 - ps, kill (later to identify and kill run-away processes)
 - icc and ifort for compiling
- Slurm (scheduler for the queue)
 - **sbatch**, **scancel** submit and cancel a job
 - **showq -l -u** show your jobs
 - **Showq -l** show all jobs
 - **-A ...** Account name

How to get around on a TACC System Frontera

- When you ssh into Frontera you are
 - on the login node (login1, or login2, or login3, etc.)
 - Login nodes are shared and used for editing files, compiling, file transfer, etc
 - Users **must** not run their executables on the login nodes
 - Correct timings can not be obtained on the login nodes
- Use `idev` to start an interactive session
- **`idev -A ...`**

General guidelines for you

- Edit files and compile either on login node or within **`idev`** session
- Running your executable: only within **`idev`** session

How to get around on a TACC System Frontera

- How to compile and execute

```
icc -xhost -O2 source.c
```

```
icpc -xhost -O2 source.c
```

```
ifort -xhost -O2 source.f90
```

```
./a.out
```

icc/icpc/ifort

Intel compiler for C/C++ and Fortran

-xhost

Optimization flag for the node architecture

-O2

General optimization flag

Multidimensional arrays in C

- There are at least three different ways of creating a 2d array in C

Multidimensional arrays in C

- There are (at least) 3 different ways to create a 2d array in C
 1. Row-by-row: multiple malloc calls
 2. Whole matrix + a vector that points to the beginning of each row
 3. Whole matrix with manual index calculation
 - Square matrix `a` with $(n \times n)$ elements
 - Accessing array element `a(i,j)` \rightarrow `a[i*n+ j]`
- Array should be allocated contiguously
 - one malloc call \rightarrow variants 2 and 3
- In your C/C++ code please use variant 3
- In your Fortran code you will ‘automatically’ use variant 3